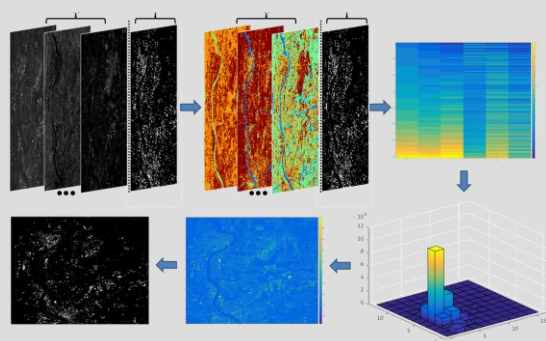


# MASADA User Guide

**Version 1.3**

Panagiotis Politis  
Christina Corbane  
Luca Maffenini  
Thomas Kemper  
Martino Pesaresi

2017



This publication is a Technical report by the Joint Research Centre (JRC), the European Commission's science and knowledge service. It aims to provide evidence-based scientific support to the European policymaking process. The scientific output expressed does not imply a policy position of the European Commission. Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use that might be made of this publication.

**Contact information**

Name: Christina Corbane

Address: European Commission, Joint Research Centre, Space, Security and Migration (Ispra), Disaster Risk Management (JRC.E.1)

E-mail: [christina.corban@ec.europa.eu](mailto:christina.corban@ec.europa.eu)

Tel.: +39 0332 78 3545

**JRC Science Hub**

<https://ec.europa.eu/jrc>

EUR 28609 EN

PDF      ISBN 978-92-79-68865-2      ISSN 1831-9424      doi:10.2760/522941

Luxembourg: Publications Office of the European Union, 2017

© European Union, 2017

The reuse of the document is authorised, provided the source is acknowledged and the original meaning or message of the texts are not distorted. The European Commission shall not be held liable for any consequences stemming from the reuse.

How to cite: Politis P; Corbane C; Maffenini L; Kemper T; Pesaresi M. MASADA USER GUIDE. EUR 28609 EN. Luxembourg (Luxembourg): Publications Office of the European Union; 2017. JRC106667

All images © European Union 2017

*Printed in Italy*

## Table of contents

Abstract .....	3
1. Introduction .....	4
1.1 Overview .....	4
1.2 Rationale .....	4
1.3 History and versioning .....	4
1.4 Main features of the MASADA tool .....	4
2. Getting started .....	7
2.1 System Requirements .....	7
2.2 Installing MASADA .....	7
2.2.1 Installing Matlab Runtime .....	7
2.2.2 Installing MapServer (optional) .....	7
2.2.3 Installing MASADA .....	8
2.2.4 Uninstalling MASADA .....	8
3. The workflow .....	9
3.1 The Symbolic Machine Learning (SML) .....	9
3.2 Textural features – Pantex .....	9
3.3 Morphological features (Characteristic-Saliency-Leveling) .....	10
3.4 Processing steps .....	11
4. Using the tool .....	22
4.1 Basic usage .....	22
4.1.1 Launching the tool .....	22
4.1.2 The graphical user interface .....	23
4.2 Output files and intermediate results .....	24
4.3 Customized table input .....	25
References .....	26
ANNEX 1. Index of Parameters .....	27
ANNEX 2. Format of parameters .....	28

# Abstract

This user guide accompanies the MASADA tool which is a public tool for the detection of built-up areas from remote sensing data.

MASADA stands for Massive Spatial Automatic Data Analytics. It has been developed in the frame of the “Global Human Settlement Layer” (GHSL) project of the European Commission’s Joint Research Centre, with the overall objective to support the production of settlement layers at regional scale, by processing high and very high resolution satellite imagery. The tool builds on the Symbolic Machine Learning (SML) classifier; a supervised classification method of remotely sensed data which allows extracting built-up information using a coarse resolution settlement map or a land cover information for learning the classifier. The image classification workflow incorporates radiometric, textural and morphological features as inputs for information extraction. Though being originally developed for built-up areas extraction, the SML classifier is a multi-purpose classifier that can be used for general land cover mapping provided there is an appropriate training data set.

The tool supports several types of multispectral optical imagery. It includes ready-to-use workflows for specific sensors, but at the same time, it allows the parametrization and customization of the workflow by the user.

Currently it includes predefined workflows for SPOT-5, SPOT-6/7, RapidEye and CBERS-4, but it was also tested with various high and very high resolution sensors like GeoEye-1, WorldView-2/3, Pléiades and Quickbird.

The software is designed for the processing of single scenes but also for batch processing of large data sets.

This user guide is a comprehensive guide to all aspects of using the MASADA tool. It includes instructions for the set-up of the software, the use of the tool and the manipulation of the data. It presents briefly the basic principles and background information on the different algorithms and modules of the methodology. Some guidelines on the parametrization of the modules are also provided and the inclusion of additional features in the sensor-oriented workflows.

# 1. Introduction

## 1.1 Overview

The MASADA tool has been developed under the [GHSL project](#) of the European Commission's [Joint Research Centre](#) with the overall objective to support the production of settlement layers by processing high and very high-resolution satellite imagery.

It builds on already implemented algorithms, carried out at the first stage of the project (texture analysis with Pantex (Pesaresi et al., 2008) and morphological decomposition with CSL (Gueguen et al., 2011; Pesaresi et al., 2012) and a novel supervised classification method, applied for the first time in the production of the Landsat multi-temporal GHSL (Pesaresi et al., 2016c, 2016b).

The purpose of the MASADA tool is to foster reproducibility of the GHSL results through the sharing of in-house analytical methods and through a handy tool for image classification. This free software policy aims at decreasing the information production cost while increasing interoperability of the information products and facilitating uptake by Member States and other stakeholders.

The objective of this documentation is to provide an in-depth walk-through this tool for massive data analytics.

## 1.2 Rationale

The access to up-to-date information on human settlements is crucial for any kind of spatial analysis related to human and physical exposure to threats such as natural disasters and conflicts or environmental contamination and degradation, to the impact of human activities on ecosystems and to the human access to resources. Monitoring of the human-induced changes, urban pressure and spatial-temporary analysis of refugee/IDP camps are crucial for effective crisis management. Additionally, several indicators, related to the above, can be derived from the information on human settlements. Data from satellite imagery and new technologies for processing earth observation data facilitate the study of population distribution. Earth observation is a unique source of information for deriving globally-consistent and evidence-based indicators for improved accountability and coherence of international frameworks.

A tool supporting the precise location and identification of settlement features from various commercial and free optical image data over large areas in different time periods, is a step forward to this direction.

## 1.3 History and versioning

The MASADA tool was initially developed in the early 2016, in the frame of the regional cooperation with the South African National Space Agency (SANSA) and the Brazil's National Institute for Space Research (Instituto Nacional de Pesquisas Espaciais, INPE), for the production of regional, multi-temporal, human settlement layers, making use of the available high resolution image datasets.

In October of 2016, these two tools were merged into one version (v.1.1), supporting different optical sensors, together with new features. Version 1.2 was released one month later to fix some bugs. The current version 1.3 was released in April 2017. It includes several improvements and optimization of the workflow.

## 1.4 Main features of the MASADA tool

The tool supports high and very high resolution optical satellite image data (between 0.5 and 10 m spatial resolution). It requires a multispectral (multi-channel) imagery, regardless of its radiometric and spatial resolutions. The presence of a higher resolution

panchromatic image (in case of bundle data) is not required, although is an asset, as it contributes to the more precise and detailed extraction of the morphological and textural image features and consequently to the better identification of the desired target information.

- *Pansharpened data*

The methodology has also been successfully applied to Pansharpened data, giving meaningful results. However, Pansharpened image data may introduce artifacts to the classification. Image fusion techniques include complex Pansharpening algorithms and local filtering for visual enhancement. From the point of view of automatic image information extraction, the above are adding noise to the original image data. Thus, they have to be modelled as black-box by the classifier.

- *Defining the parameters*

To support this openness to every multispectral optical sensor and for modularization and tuning purposes, the parametrization is not hard-coded; on the contrary, the input parameters could be accessed by the user through an .ods (OpenDocument Spreadsheet that can be edited from any spreadsheet application like Microsoft Excel, LibreOffice, Calc or Openoffice Calc)formatted file or a spreadsheet (.xls or .csv) table sheet. Some .ods example files are also included in the installation package with the proposed parameters. The user is encouraged to tune the algorithm, having access to the parameters. Provided that, there is also the ability of modularization of the presented workflow by selecting different parametrization, in order to achieve optimal results, depending on the image type, the landscape, the settlement pattern and density, seasonal effects linked to the acquisition period and – of course – the desirable output.

- *Intermediate results and batch processing*

The tool supports single image processing, as well as batch processing. Intermediate results can be exported in a test phase for optimizing the parameters for the batch processing. With that way the user can check the intermediate results and understand how they affect the output product by detecting anomalies in the final results. After deciding the proper set of parameters, a batch mode processing can be applied for massive data run.

- *Training phase*

The supervised classification is based on a training set. This dataset should not be sample-based, but cover the whole area of interest. Any spatial information describing the presence of built-up area, like building footprint data or generalized data like land use / land cover information layer can be used in the learning phase of the classification.

- *Validation phase*

The presence of a validation set is optional, but is recommended for the evaluation of the output and for the cross-comparison of the parametrization. In this case, a confusion map and performance statistics are calculated and saved in .tiff and csv formats respectively.

- *Output files*

The output files consist of GEOTIFF images of the same input resolution of the multispectral image and with the same geoinformation. They include the final classified output and the confidence layer, accompanied by three CSV files comprising the I/O parameterization, execution times-control conditions, general metadata and performance metrics.

To sum up, this version 1.3 of the tool includes the below characteristics:

- *Workflow oriented to settlement identification. However, other it is possible to extract other land cover classes (one class per each run) using a proper training. In that case, the SML can be applied on the radiometric stack only. Multi-sensor support. Prerequisite is a high or very high resolution optical multispectral image. Bundle data is supported.*
- *Tailored workflows for specific products (SPOT-5 bundle data, SPOT 6&7 bundle data, RapidEye 3A, CBERS-4 PAN/MUX).*
- *Ability to process other bulk optical image data from custom mode.*
- *Flexibility to manage the parameters manually per scene and per sensor through a table format (.ods).*
- *Selection of learning set, validation set and other auxiliary layers through map services or local image files.*
- *Features selection (extracted features, radiometric bands) to participate in the SML analysis and further parametrization.*
- *Image enhancement for the feature extraction.*
- *Option to export intermediate results.*
- *Options to define the method for the classification of the confidence layers.*
- *Single scene or batch processing.*

This user manual provides instructions about the installation of the required software and the data set-up (chapter 2). The classifier and the basic aspects and definitions are illustrated in the chapter 3 along with the workflow schema. The graphical user interface, the configuration file and the use of the tool are presented in chapter 4.

## 2. Getting started

The algorithms have been coded in the scripting language of MATLAB R2016b. We note here that some sub-routines have been developed in C and JAVA programming languages in order to resolve performance issues and improve the modules interfacing. For image reading and writing, we employ MATLAB wrappers of the respective GDAL functionalities (Geospatial Data Abstraction Library: <http://www.gdal.org/>).

### 2.1 System Requirements

To use MASADA, your system must meet the following minimum requirements:

Operating system	Processor	Disk space	RAM
Windows 7 or Windows 10	Any Intel or AMD x86-64 processor	100 Mb (+ 700 Mb for Matlab Runtime)	16 GB (recommended)

### 2.2 Installing MASADA

To have MASADA running on your computer you need to install also this components:

1. Matlab Runtime
2. MapServer (optional)

#### 2.2.1 Installing Matlab Runtime

Verify the MATLAB Runtime is installed and ensure you have installed version **9.1 (R2016b)**.

If not already installed, download the Windows 64-bit R2016b version of the MATLAB Runtime from the MathWorks Web site:

<http://www.mathworks.com/products/compiler/mcr/index.html>

For more information about the MATLAB Runtime and the MATLAB Runtime installer, see Package and Distribute in the MATLAB Compiler documentation in the MathWorks Documentation Center.

#### 2.2.2 Installing MapServer (optional)

MASADA can work with WMS services to fetch reference data. These services can be online or can be provided by the user using MapServer WMS.

MapServer is an [Open Source](#) platform for publishing spatial data and interactive mapping applications to the web.

Originally developed in the mid-1990's at the University of Minnesota, MapServer is released under an [MIT-style license](#), and runs on all major platforms ([Windows](#), [Linux](#), [Mac OS X](#)).



It support numerous [Open Geospatial Consortium](http://opengeospatial.org/) (OGC) standards like **WMS** and **WCS**.

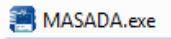
For instructions about installation and configuration in your machine or local infrastructure check the official MapServer documentations:

[http://mapserver.org/ogc/wms\\_server.html#setting-up-a-wms-server-using-mapserver](http://mapserver.org/ogc/wms_server.html#setting-up-a-wms-server-using-mapserver)

### **2.2.3 Installing MASADA**

The Windows installer is included in the zip file.

After saving it on your hard drive, unzip the zip file and run the executable directly:



### **2.2.4 Uninstalling MASADA**

To remove MASADA just delete the executable file and its folder.

Matlab Runtime can be uninstalled as well if not needed by other applications.

# 3. The workflow

The MASADA tool builds on the Symbolic Machine Learning classifier with two additional components for enhanced feature extraction: the Pantex and the CSL. Those three modules represent the building block of the workflow. They are briefly described in the following sections:

## 3.1 The Symbolic Machine Learning (SML)

The technology at the core of the GHSL relies on the Symbolic Machine Learning (SML) supervised classifier (Pesaresi et al., 2016c).

The SML schema is based on two relatively independent steps:

1. Reduce the data instances to a symbolic representation (unique discrete data-sequences);
2. Evaluate the association between the unique data-sequences subdivided into two parts: X (input features) and Y (known class abstraction).

The data-abstraction association is evaluated by a confidence measure called ENDI (Evidence-based Normalized Differential Index) which is produced in the continuous  $[-1, 1]$  range. To address the need for crisp classification (i.e. built-up and non-built-up), a threshold is estimated for assigning each data sequence to a single class. Details on the SML algorithm and its eligibility in the framework of big data analytics may be found in (Pesaresi et al., 2016c, 2016d). The application of the SML classifier to the processing of Landsat data records of the past 40 years is discussed in (Pesaresi et al., 2016b) and its deployment to the classification of both Sentinel-1 and Sentinel-2 data is presented in (Pesaresi et al., 2016a).

In the GHSL paradigm, the built-up area class is defined as the union of all the spatial units collected by the specific sensor and containing a building or part of it. The satellite data is described by 1) multi-spectral radiometric features, 2) textural features and 3) morphological features, extracted from the multispectral and panchromatic bands.

Two in-house developed modules are available in the MASADA tool for extracting textural and morphological features as inputs to the SML. They are briefly described hereafter:

## 3.2 Textural features – Pantex

The texture-derived built-up presence index (PanTex) (Pesaresi et al., 2008) is calculated from the textural characteristics of panchromatic satellite data. The index is based on fuzzy rule-based composition of anisotropic textural co-occurrence measures derived from the satellite data by the gray-level co-occurrence matrix (GLCM). In the GLCM approach, several key parameters play a role for the textural index calculation: the selected statistics, the vector displacement, the window size and the number of gray levels in the image. All of these parameters are related to the spatial and spectral resolution of the image, and the spatial characteristics (dimension, shape) of the different classes to be detected:

- **The selected statistics:** The detection of built-up areas is based on the calculation of contrast textural measure derived from the GLCM.
- **The vector of displacement:** in the Pantex a total of 4 combinations of distance and angle of the displacement vector have been selected inside a predefined window size (Figure 1). The anisotropic information associated with the GLCM textural measure is exploited using the minimum operator ( $\cap$  operator) of the 4 combinations instead of the usual average for integrating the different texture directions.

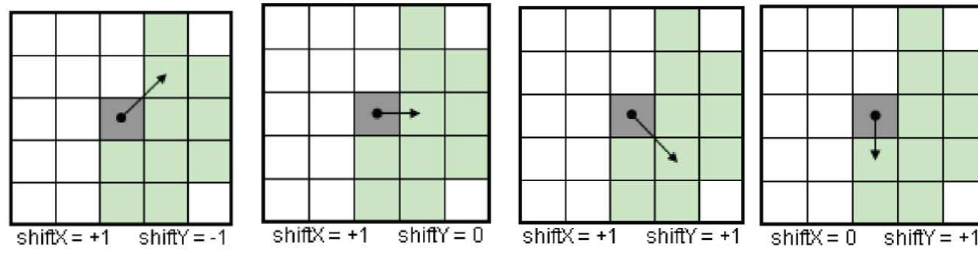


Figure 1. Four GLCM displacement vectors used in the Pantex

- **The window size:** is a parameter to be defined by the user. The value is derived by the empirical estimation of relation between window size and detection accuracy for a specific set of settlement structures. In the case of an image with a spatial resolution of 5 meters, the target minimal settlement structure detectable is composed by at least two buildings, some open spaces, and roads in between with an estimated minimal footprint of approximately 50 m, corresponding to a window size of 9 x 9 pixels at the given pixel size of 5 m.

### 3.3 Morphological features (Characteristic-Saliency-Leveling)

The morphological image features are based on the characteristic-saliency-level (CSL) of the multi-scale morphological decomposition schema (Pesaresi et al., 2012). The latter is a multi-resolution pyramid transform of the image defined by series of morphological filters (Gueguen et al., 2011). The CSL model allows the compression and storage of the multi-scale information contained morphological profiles into raster data layers, used for further analytic purposes.

The user has to define the lambda vector of scales of the multi-scale morphological decomposition usually ranging from 25 square meters until 1.5 hectares of surface. This vector of scales corresponds to the a priori knowledge about the expected size of the most common built-up structures.

In the CSL model the morphological "characteristic" **C** layer reports about the scale space image decomposition schema. It maintains separated the closing and opening image domains corresponding to the image connected components showing respectively, darker and brighter grey level values with respect to the surrounding image components.

The "saliency" **S** layer reports about the amount of image contrast explained by the characteristic scale detected in the **C** layer. Consequently, it can be used as an estimation of the saliency of the detected image connected component with respect to the relative background.

The "levelling" or **L** layer reports about the absolute spectral characteristics (contrast) of the image information. It corresponds to the remaining signal after the filtering (Figure 2).

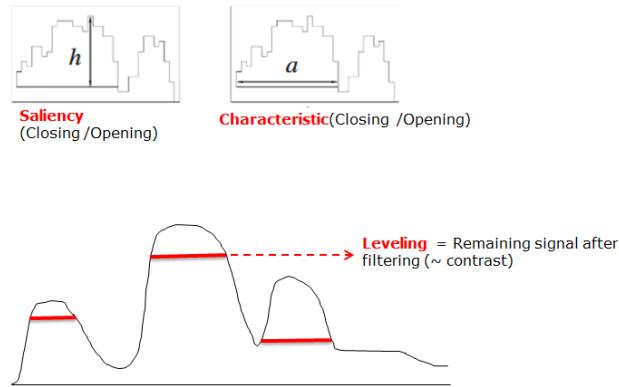


Figure 2. Simplified illustration of the components of the CSL model: Characteristic, Saliency and Levelling.

### 3.4 Processing steps

The workflow is divided practically into 10 processing steps each of them can be parameterized with at least one parameter. The following two figures (Figure 3 and Figure 4) provide an overview of the workflow and the set of parameters associated with each step. The parameters can be easily defined/modified by the user in the .ods external file which is loaded through the user interface at the beginning of the process (see section 4.1.2)

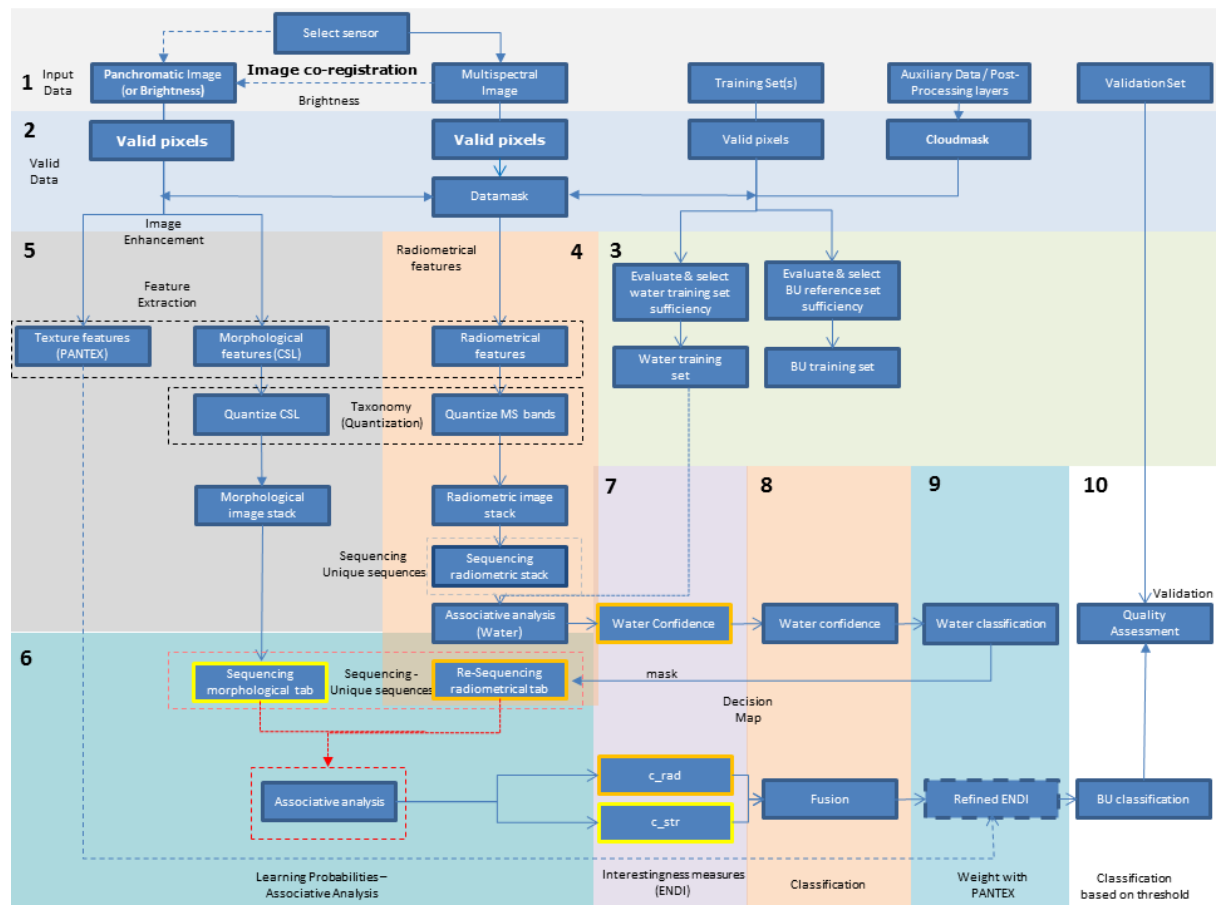


Figure 3. Image processing workflow implemented in the MASADA tool

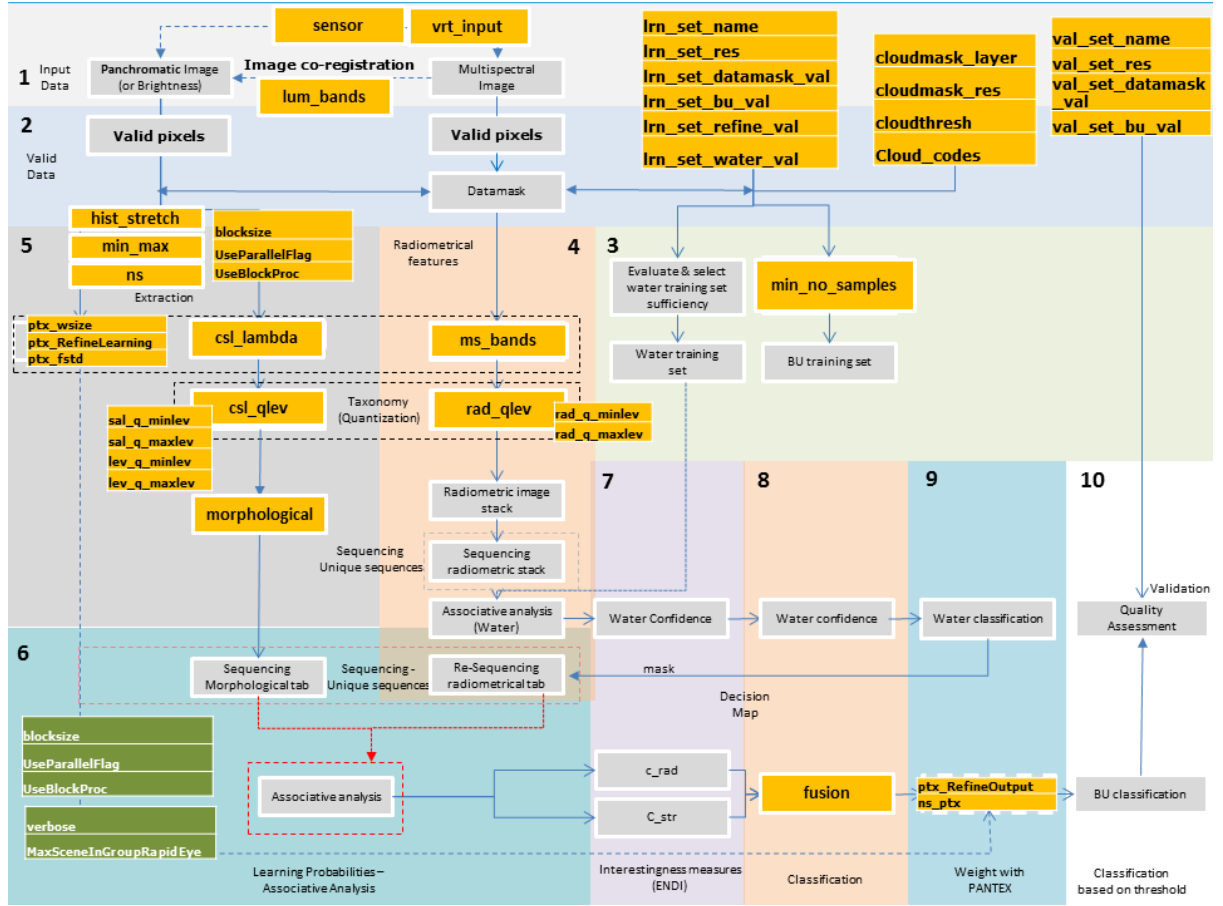


Figure 4. Set of parameters to be defined by the user in the .ods file.

In the following, each of the 10 steps is briefly introduced together with its parameters:

## 1. Input data and sensor selection

### 1.1 Read parameters

The first step consists in reading the parameter/configuration file (.ods) which is loaded at the beginning of the process in the user interface (see section 4.1.2).

The input sensor type will define the workflow:

- In case of a panchromatic band is available, the user may choose to calculate morphological and textural features on that band to enhance the detection of built-up areas.
- In case of non-availability of panchromatic band, a brightness band is calculated from the visible bands of the multispectral input data as follows:

$$\text{Brightness} = \text{Max}(\text{Visible Bands})$$

The assumption is that the built-up structures are the most shining features in the optical bands domain.

The user may then choose to calculate textural and morphological features using the Brightness band as input.

<b>Parameter</b>	<b>Definition</b>
<b>Sensor</b>	Selection of the supported sensors : SPOT-5, SPOT-6 & SPOT-7, RapidEye, CBERS-4 PAN/MUX, Custom
<b>vrt_input</b>	Selection of a .vrt file instead of single .tiff images. In that case the full path to the .vrt file should be specified.  The MASADA tool performs an automatic image co-registration to align the multispectral and the panchromatic image bands. This process is not visible to the user and cannot be parametrized.
<b>lum_bands</b>	Selection of Multispectral or Pansharpened image bands for <i>Brightness</i> computation (this is defined in case of non-availability of a Panchromatic band).

## 1.2 Defining input training and validation sets

The input data used for learning the SML classifier has to be defined at the first stage in the .ods file.

Optionally a validation set may be also used for computing the confusion matrix map and associate performance metrics summarizing the quality of the classification.

A cloud mask may be also defined for masking purposes. It may consist of a .gml file or any other cloud mask defined by the user.

<b>Parameter</b>	<b>Definition</b>
<b>lrn_set_name</b>	Full path to the training/learning set. It could correspond to a layer name served through a WMS (when a Mapserver is installed) or to local file (.tiff or a .vrt).  The learning can correspond to any thematic map (not necessarily binary classification) of better or lower resolution than the input image but preferably covering the full extent of the input image under processing. If the learning data covers only a portion of the input image, then the processed area of interest will be adjusted automatically to the intersection between the input image and the learning data.  In case the geographical extent of the learning exceeds that of the input image, the learning will be clipped to the same extent of the input data.
<b>lrn_set_res</b>	Pixel size in meters of the training set in case the data is served through a WMS.  If the input training set is a raster image (.tiff or a

	.vrt file), the pixel size will be defined automatically.
<b>lrn_set_datamask_val</b>	Vector of digital codes in the learning set that correspond to valid data. No Data values should be excluded from the vector of valid data.
<b>lrn_set_bu_val</b>	Digital codes in the learning set that correspond to the positive examples: e.g. corresponding to built-up areas.
<b>lrn_set_refine_val</b>	Digital codes in the references that are positive built-up areas examples but that require a refinement with the Pantex textural feature. This may be useful in case of land cover class which includes scattered buildings (for instance smallholdings etc.)
<b>lrn_set_water_val</b>	Digital codes in the learning set that are positive water areas examples. In case the user may want to extract also the water class with the SML.
<b>cloudmask_layer</b>	Full path to ancillary cloudmask layer (.gml , udm, .tiff, .jp2, .vrt) or WMS layer name. If empty no cloudmask will be requested. If "draft" an approximate cloudmask is computed based on the bright pixel values (cutoff value of > 95% of the visible bands range) and morphological operators.
<b>cloudmask_res</b>	Pixel size (in meters) of the cloudmask layer in case it is served through a WMS. If raster image, the pixel size will be determined automatically.
<b>cloudthresh</b>	Cutoff values for bright objects from the RGB bands in case the user selected to calculate a rough cloudmask.
<b>Cloud_codes</b>	Digital code corresponding to clouds for masking purposes.
<b>val_set_name</b>	<p>Full path to the validation set. It could correspond to a layer named served through a WMS or local file (.tiff or a .vrt)</p> <p>The validation data can correspond to any thematic map (not necessarily binary classification) covering partially or fully the extent of the input image.</p> <p>In case the geographical extent of the validation exceeds that of the input image, the validation will be clipped to the same extent of the input data.</p> <p>In case, the geographical extent of the validation data partially covers that of the input image, the metrics will be calculated only for the extent where the validation set is provided corresponding to the the intersection with the valid image data.</p>

<b>val_set_res</b>	Pixel size in meters of the validation set. If the input validation set is a raster image (.tiff) or a .vrt file, the pixel size will be defined automatically.
<b>val_set_datamask_val</b>	Vector of digital codes in the validation set that correspond to valid data.
<b>val_set_bu_val</b>	Digital codes in the validation set that correspond to the positive examples: e.g. corresponding to built-up areas.

### 1.3 Define workflow by sensor

Two types of workflows can be run:

- a) Considering only radiometric features,*
- b) Considering also morphological and textural features.*

The selection of the type of workflow depends on the type of sensor used and the input radiometric and spatial resolutions.

In case of very high resolution remote sensing data (better than 6 meters) , it is recommended to calculate textural and morphological features which describe the structural characteristics of the objects present in the scene and may enhance the detection of built-up areas.

<b>Parameter</b>	<b>Definition</b>
<b>morphological</b>	In case it is set to "True", morphological features will be calculated following the CSL model and an image stack called "structural stack" comprising the Characteristic, Saliency and Leveling bands will be generated for further processing in the SML.
<b>ptx_RefineOutput</b>	In case it is set to "True", the Pantex will be run and a textural feature will be calculated for refining the output confidence layer (refinement of ENDI).

## 2. Valid data computation

Given the previously defined input data, the selected parameters and workflow, the second step of the process loads the input datasets, calculates the datamask and initializes the computation of the required features. It does not require any further parametrization. The following operations are performed sequentially:

- 2.1 Calculate datamask (valid pixels from images)
- 2.2 Calculation of the rough cloud mask (in case this option has been selected in stage 1)
- 2.3 Update of the datamask with newly generated cloudmask or with the ancillary file corresponding to cloudmask.



### 3. Evaluation of training data

This step evaluates the learning data (e.g. the positive samples corresponding to water and the positive samples corresponding to built-up areas). In case of built-up samples, the user can define a minimum number of positive samples as a condition for running the SML. The default is set to 0.

3.1 Evaluate the water training set

3.2 Evaluate the built-up training set

<b>Parameter</b>	<b>Definition</b>
<b>min_no_samples</b>	Minimum number of positive built-up samples. If lower than this value, the image will be skipped since there are not enough positive samples for the learning.

### 4. Quantization – Data reduction

This stage corresponds to the data quantization- reduction step of the SML classifier (see section 3).

4.1 Quantize multispectral data

In this step, the multispectral image stack is first quantized for further processing in the SML classification.

<b>Parameter</b>	<b>Definition</b>
<b>ms_bands</b>	Radiometric bands to be included in the stack for data quantization/reduction in the SML classifier.
<b>rad_qlev</b>	Quantization parameter corresponding to then number of levels for reducing the input data. It is recommended to run several tests for defining this parameter. An optimal number of levels should yield an average support in the range $10^2$ - $10^3$ . The output value of the <b>Average Support (smlAvgSuppRad)</b> can be checked from the outputted <i>outrec.csv</i> file.
<b>rad_q_minlev</b>	Minimum cutoff value (percentiles) for rescaling the radiometric bands prior to quantization
<b>rad_q_maxlev</b>	Maximum cutoff value (percentiles) for rescaling the radiometric bands prior to quantization

4.2 Quantize morphological data

In this step, the morphological image stack is quantized for further processing in the SML classification. The morphological image stack comprises the output of the CSL model (Characteristic-Saliency-Leveling). Each morphological feature is treated separately in the data reduction phase due to the nature and data type of

the output. The **Characteristic** band is not quantized/reduced due to the small range of values obtained for this feature (usually less than 10 levels).

<b>Parameter</b>	<b>Definition</b>
<b>csl_qlev</b>	Quantization parameter corresponding to the number of levels for reducing the morphological data (or structural features). It is recommended to run several tests for defining this parameter. An optimal number of levels should yield an average support in the range $10^2$ - $10^3$ .  The output value of the <b>Average Support (smlAvgSuppStruc)</b> can be checked from the outputted <i>outrec.csv</i> file.
<b>sal_q_minlev</b>	Minimum cutoff value (percentiles) for rescaling the <b>Saliency</b> band prior to quantization
<b>sal_q_maxlev</b>	Maximum cutoff value (percentiles) for rescaling the <b>Saliency</b> band prior to quantization
<b>lev_q_minlev</b>	Minimum cutoff value (percentiles) for rescaling the <b>Levelling</b> band prior to quantization
<b>lev_q_maxlev</b>	Maximum cutoff value (percentiles) for rescaling the <b>Levelling</b> band prior to quantization

#### 4.3 Sequencing radiometric and morphological image stacks

Following the quantization of the radiometric and morphological image stacks, the data instances are converted to a symbolic representation and two sets of unique data sequences are calculated: morphological sequences and radiometric sequences. Those unique sequences are then used in the next step of the SML Associative Analysis for calculating two distinct confidence measures (ENDI): one for derived from radiometric characteristic and one derived from structural (morphological characteristics) (see step 7 of the workflow).

## 5. Image enhancement and textural feature extraction

In this stage, some options for image enhancement are proposed for improving the contrast of the panchromatic band or the calculated Brightness prior to the extraction of textural features and morphological features. This step is recommended in the user wishes to include textural features and morphological features in the workflow.

### 5.1 Image Enhancement

Two different approaches for image enhancement are proposed: a) histogram stretching and b) standardization which corresponds to a rescaling of the image between the following two values:

$$[Mean-Ns * Std ; Mean+NS * Std]$$

The *Mean* and the Standard deviation of the image values *Std* are calculated in the learning set domain. *Ns* corresponds to the number of standard deviations and is set by the user.

<b>Parameter</b>	<b>Definition</b>
<b>hist_stretch</b>	Selection of the image enhancement method: <ul style="list-style-type: none"> <li>- histogram stretching</li> <li>- standardization rescaling</li> <li>- if null the image will not be rescaled</li> </ul>
<b>min_max</b>	In case the histogram stretching method is selected, this parameter corresponds to the contrast limits of the input image, specified as a two-element numeric vector with values between 0 and 1. Default is the bottom 1% (min) and the top 1% (max) of all pixel values.
<b>ns</b>	Number of standard deviations to be considered in standardization process if the standardization rescaling method is selected.

## 5.2 Calculate textural features (Pantex)

The textural feature is calculated using the Pantex algorithm (see section 3.2). The calculation of the textural feature requires the setting of window size and selecting the option of refinement of the learning set. The latter is recommended in few cases of scattered settlements in the training set that may introduce noise in the SML classifier.

<b>Parameter</b>	<b>Definition</b>
<b>ptx_wsize</b>	Window (kernel) size for Pantex calculation (expressed in meters)
<b>ptx_RefineLearning</b>	Selection of the option of refinement of the learning set using the textural feature.
<b>ptx_fstd</b>	Standard deviation factor for cutoff and masking during the refinement of the learning set.

## 5.3 Calculate Morphological features (Characteristic-Saliency-Leveling: CSL)

The morphological model is calculated following the CSL model. The user needs to define the lambda vector of scales of the multi-scale morphological decomposition usually ranging from 25 square meters until 1.5 hectares of surface. This vector of scales corresponds to the a priori knowledge about the expected size of the most common built-up structures.

<b>Parameter</b>	<b>Definition</b>
<b>csl_lambda</b>	The lambda vector of scales of the multi-scale morphological decomposition expressed in meters. See Annex 2 for details on how to enter vector parameters in the configuration file.

## 6. Associative analysis

This is the core step of the SML classifier where the relationship between the unique morphological and/or radiometric sequences and the learning set is calculated. The Associative Analysis quantifies the frequencies of occurrence of the unique sequences in the positive and negative samples (Pesaresi et al., 2016c, 2016d).

<b>Parameter</b>	<b>Definition</b>
<b>blocksize</b>	block size for block processing
<b>UseParallelFlag</b>	Enable parallel mode on machine with multicore processors. When set to true, block processing attempts to run in parallel mode, distributing the processing across multiple workers (MATLAB® sessions) in an open MATLAB pool.
<b>UseBlockProc</b>	Selection of block processing option. It processes images by block, reducing the amount of memory required.
<b>verbose</b>	Select the option to save intermediate files
<b>MaxSceneInGroupRapidEye</b>	Select the option of extending the processing to neighbouring scenes by defining the number of adjacent tiles (warning: might be very heavy computationally) only for RapidEye Level 3A product (25x25km tiles)

## 7. Confidence Measure (ENDI)

The results of the Associative Analysis are represented in terms of a confidence measure: Evidence-based normalized differential index (ENDI) which is a generalization of a measure from the confidence family to the case in which a set of positive/negative not-mutually-exclusive examples is provided.

$$\Phi_E^a(X, Y^+, Y^-) = \frac{f_{pos-} - f_{neg}}{f_{pos+} + f_{neg}}$$

$$\Phi_E^b(X, Y^+, Y^-) = \frac{p_{pos} - p_{neg}}{p_{pos} + p_{neg}}$$

where  $f_{pos}$  and  $f_{neg}$  are the frequencies of the joint occurrences among  $X$  data instances (unique data sequences) and the positive and negative learning instances respectively ( $Y^+$  and  $Y^-$ );  $p_{pos}$  and  $p_{neg}$  represent the empirical probabilities calculated as  $p_{pos} = \frac{f_{pos}}{N_{pos}}$ ;  $p_{neg} = \frac{f_{neg}}{N_{neg}}$ ;  $N_{pos}$  and  $N_{neg}$  denote the total of the positive and negative samples of the training set, respectively. The ENDI measure of confidence has values ranging between  $[-1, +1]$ .

In case morphological features are calculated in the workflow, two ENDI measures will be obtained in this phase: a measure of confidence for radiometric features ( $c_{rad}$ ) and a measure of confidence for morphological features ( $c_{str}$ ).

In case the user selects the option to classify water, a water confidence layer is also generated.

## 8. Classification

At the classification stage, the ENDI measures of confidence are thresholded to obtain a binary map (e.g. built-up, non-built-up). An automatic thresholding method based on the OTSU algorithm is implemented in the MASADA workflow (Huang and Wang, 2009). This global image threshold approach chooses the threshold to minimize the intraclass variance of the black (e.g. non-built-up) and white pixels (e.g. built-up).

Given that in some situations two ENDI measures are available (i.e a measure of confidence for radiometric features ( $c_{rad}$ ) and a measure of confidence for morphological features ( $c_{str}$ )), several options are offered to the users for combining the two outputs into one single binary classification as follows:

<b>Parameter</b>	<b>Definition</b>
<b>fusion</b>	<p>Fusion operators for combining the ENDI measures:</p> <ul style="list-style-type: none"> <li>- 'mean': average of <math>c_{rad}[-1, 1]</math> and <math>c_{str}[-1, 1]</math></li> <li>- 'max': maximum of <math>c_{rad}[-1, 1]</math> and <math>c_{str}[-1, 1]</math></li> <li>- 'min': minimum of <math>c_{rad}[-1, 1]</math> and <math>c_{str}[-1, 1]</math></li> <li>- 'intersection': intersection (minimum) of binarized confidence measures : <math>c_{rad}_{\{0,1\}} \cap c_{str}_{\{0,1\}}</math></li> <li>- 'union': union (maximum) of binarized confidence measures : <math>c_{rad}_{\{0,1\}} \cup c_{str}_{\{0,1\}}</math></li> </ul>

## 9. Refinement of output

In some cases, using only the radiometric and morphological features may result in over detection of roads and open spaces and agricultural fields in the built-up class. To overcome this issue, it is suggested to refine the output confidence using the calculated textural feature (Pantex). This is an optional step that has to be decided at the first stage of the process prior to running the workflow. Two parameters allow controlling this step:

<b>Parameter</b>	<b>Definition</b>
<b>ptx_RefineOutput</b>	If set to true, the output confidence will be refined using the textural feature.
<b>ns_ptx</b>	Parameter vector for rescaling the Pantex output in the range 0-1. The larger the value, the fewer the influence of the textural on the refined output (in case the user decides to have a less conservative to avoid large omissions).

## 10. Validation

In case a validation set is available and defined in the input parameter file (.ods) a set of quality metrics are calculated and a confusion map is generated as part of the outputs. Those metrics are derived from the following confusion matrix:

*Table 1. Illustration of a two -class (built-up , non built-up) confusion matrix*

	Reference (actual, observation) positive (BU =1 )	Reference (actual, observation) negative (BU =0 )	
Prediction (classification, label) positive (BU =1 )	True Positive (TP) Hit a	False Positive (FP) (Commission) b	Predicted Positive (PP = TP+FP)
Prediction (classification, label) negative (BU =0 )	False Negative (FN) (Omissions) c	True Negative (TN) correct rejection d	Predicted Negative (PN = FN+TN)
	Real Positive (RP = TP+FN)	Real Negative (RN = FP+TN)	Total population (N = TP+FP+FN+TN)

### 10.1 Confusion map

The confusion map summarizes the following four statistics into one image with five distinct values:

- 0 → No data
- 1 → True Negative (TN)
- 2 → Omission error (FN)
- 3 → Commission error (FP)
- 4 → True Positive (TP)

## 10.2 Performance Metrics

Several additional statistics are derived from the confusion matrix. They are summarized in the output **Performance\_Metrics.csv** file. The main statistics that can be used for comparing different output and for fine-tuning the workflow are the following:

$$\text{Accuracy} = \frac{TP+TN}{N}$$

$$\text{Balanced Accuracy} = \frac{TPR+TNR}{2} = \frac{1}{2} \left( \frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right)$$

$$\text{Omission error} = \frac{FN}{FN+TP}$$

$$\text{Commission error} = \frac{FP}{FP+TP}$$

$$\text{Sensitivity (True Positive Rate)} = \frac{TP}{TP+FN}$$

$$\text{Specificity (True Negative Rate)} = \frac{TN}{TN+FP}$$

$$\text{Kappa} = \frac{2(TP \times TN - FP \times FN)}{(TP+FN)(FN+TN) + (TP+FP)(FP+TN)}$$

$$\text{Informedness} = \frac{TP \times TN - FP \times FN}{(TP+FN)(FP+TN)}$$

# 4. Using the tool

## 4.1 Basic usage

### 4.1.1 Launching the tool

There are two ways to initialize the tool a. by clicking on the executable MASADA.exe (see section 2.2.3), b. by typing the full path to the executable in the Windows Command Prompt (cmd.exe) and pressing ENTER (Figure 5).

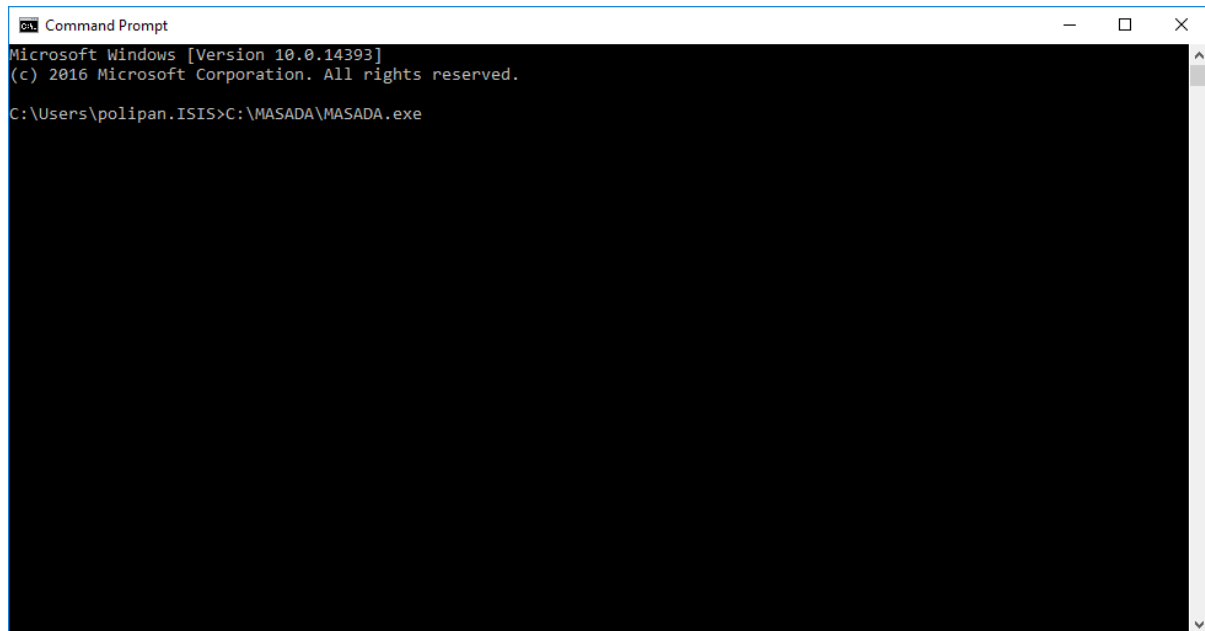


Figure 5. Running the MASADA executable from windows command prompt

The graphical user interface (GUI) will be loaded (Figure 6). During the execution of the tool, informative messages about the progress will be shown in the Command Prompt window. If the initialization is done directly by clicking the .exe file, the progress

messages and the error logs, are monitored in the log file "GHSL\_process\_status.txt". This file is saved inside output folder (as defined by the user) along with the timestamp, the identifiers of the input image and the processing status.

### 4.1.2 The graphical user interface

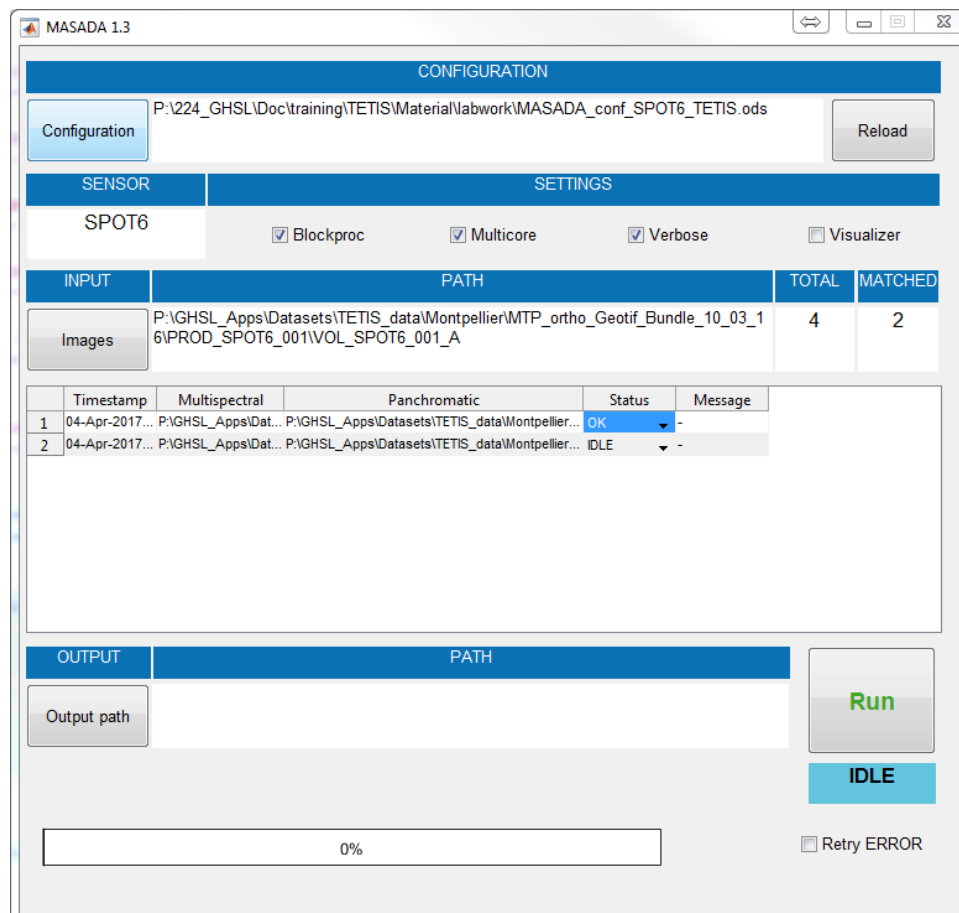


Figure 6. Graphical user interface of the MASADA tool

The graphical user interface is structured as follows:

Firstly, the configuration file containing the defined parameters should be loaded in the **CONFIGURATION** section. The example .ods configuration file provided with the MASADA tool could be used as a basis for the parametrization.

The list of parameters that need to be tuned to the input sensor data is provided in [Annex 1](#). The format and data type of the parameters are explained in [Annex 2](#) together with some examples.

In the section **SENSOR** the user sees one of the predefined specific workflows, namely: a. SPOT-5 bundled data, b. SPOT-6 bundled data, c. SPOT-7 bundled data, d. RapidEye 3A, e. CBERS-4 PANMUX, or the "Custom" option for customizing the workflow and applying it in bulk data.

If the option "Custom" is pointed, then the pop-up window will ask for a spreadsheet file to be selected (refer to the example of Multi\_Sensors\_List.xlsx file provided with the tool for an example of a list of images to be processed-see section 4.3).



In the **INPUT** section, the path to the parent folder containing the input images (Panchromatic (P) and Multispectral (MS)) should be specified. The input images can be in the form of .tiff files or .vrt files.

The list of automatically matched multispectral and panchromatic images will then appear.

Under Status, it is possible to check whether:

- an image has been already processed or should be excluded from the processing (**Status: OK**),
- is being processed (**Status: running**),
- generated an error during the process (**Status: error**),
- needs to be processed (**Status: IDLE**).

In the **OUTPUT** section, the user needs to define the output path where all the outputs and intermediate results will be saved.

## 4.2 Output files and intermediate results

It is useful to export both the main output results but also the intermediate files of the processing in order to examine stepwise the results and to fine-tune accordingly the parameters. The output and intermediate results correspond to the following files:

### Output files

- 1.1 **BU\_class.tif** → the classified output (0: noData, 1: No built-up, 2: water, 255: built-up)
- 1.2 **BU\_Cnfd.tif** → ENDI confidence layer (with the suffix \_Min \_Mean \_Max depending on the selected fusion method)
- 1.3 **Metadata.csv** → metadata of the processing: input image paths, geoinformation, reference layer for BU and water, count of reference pixels, count of detected pixels per class, BU density, confusion matrix, selected options, etc.)
- 1.4 **Outrec.csv** → data regarding the processing, total processing time, processing time per each module, levels, quantization, image statistics)
- 1.5 **Perfomance\_Metrics.csv** → accuracy metrics derived from confusion matrix (if validation set is provided)

### Intermediate results

- 1.6 **BULearning.tif** → learning set
- 1.7 **BUValidation.tif** → validation set
- 1.8 **c\_rad.tif** → ENDI confidence of radiometric stack
- 1.9 **c\_str.tif** → ENDI confidence of structural stack
- 1.10 **c\_water.tif** → ENDI confidence of water
- 1.11 **watermask.tif** → binary watermask
- 1.12 **cloudmask.tif** → cloudmask
- 1.13 **csI.tif** → 5bands CSL
- 1.14 **ptx.tif** → Pantex
- 1.15 **datamask.tif** → Datamask calculated and applied in the workflow

1.16 **Confusion\_map.tif** → error map derived from confusion matrix including: True Positives, False Positives, True Negatives (Commission errors), False Negatives (Omission errors)

### 4.3 Customized table input

As a way to support the processing any type of sensor and for facilitating the experimental mode, the option of inserting the inputs and the parameters through a customized table is available. The parameters can be replaced in a spreadsheet file format, for example excel or csv file (**see example of Multi\_Sensors\_List.xlsx file provided with the tool for an example of a list of images to be processed**).

Some basic rules should be considered:

- First step is to select the "Custom" value for sensor parameter in the configuration file (.ods).
- Second step is to load the configuration file in the GUI, then load the customized table (e.g. **Multi\_Sensors\_List.xlsx**) under the input Images button of the GUI.
- It is required to include a header on the top of the customized table with the basic 3 variables (columns), named:
  1. MS: full path of the multispectral image.
  2. PAN: full path of the panchromatic image.
  3. SENSOR: name of the sensor (for the available workflows).
- In the next columns the user has the option to add the name of the variable(s) to be considered. Note that the parameter name should be exactly the same as in the .ods file and it is case sensitive.
- If the variable name is not compatible, the parameter of the .ods file will be used.
- If both variable names are not valid, an error message will be appeared during the processing phase.
- The variable value defined in the customized file has priority over the one in the .ods file.

The advantages of using the customized table input are:

- It is a more readable format for simple users
- It supports any MS image sensor and
- It allows to define directly the input image(s)
- It allows to process different sensor products and image types within the same run; using the .ods format only same sensor images with the same parameters can be processed.
- It allows to run the same images with different parametrization by defining accordingly the parameter(s) to be changed.

# References

- Gueguen, L., Soille, P., Pesaresi, M., 2011. Structure extraction and characterization from Differential Morphological Decomposition. Presented at the The seventh conference on image information mining: Geospatial intelligence from earth observation (IIM), 2011 ESA EUSC JRC, pp. 53–58.
- Huang, D.-Y., Wang, C.-H., 2009. Optimal multi-level thresholding using a two-stage Otsu optimization approach. *Pattern Recognit. Lett.* 30, 275–284. doi:10.1016/j.patrec.2008.10.003
- Pesaresi, M., Corbane, C., Julea, A., Florczyk, A., Syrris, V., Soille, P., 2016a. Assessment of the Added-Value of Sentinel-2 for Detecting Built-up Areas. *Remote Sens.* 8, 299. doi:10.3390/rs8040299
- Pesaresi, M., Ehrlich, D., Ferri, S., Florczyk, A., Carneiro Freire Sergio, M., Halkia, S., Julea, A., Kemper, T., Soille, P., Syrris, V., 2016b. Operating procedure for the production of the Global Human Settlement Layer from Landsat data of the epochs 1975, 1990, 2000, and 2014. Publications Office of the European Union.
- Pesaresi, M., Gerhardinger, A., Kayitakire, F., 2008. A Robust Built-up Area Presence Index by Anisotropic Rotation-Invariant Textural Measure. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 1, 180–192.
- Pesaresi, M., Ouzounis, G.K., Gueguen, L., 2012. A new compact representation of morphological profiles: report on first massive VHR image processing at the JRC, in: Shen, S.S., Lewis, P.E. (Eds.), . pp. 839025–839025–6. doi:10.1117/12.920291
- Pesaresi, M., Syrris, V., Julea, A., 2016c. A New Method for Earth Observation Data Analytics Based on Symbolic Machine Learning. *Remote Sens.* 8, 399. doi:10.3390/rs8050399
- Pesaresi, M., Vasileios, S., Julea, A., 2016d. Analyzing big remote sensing data via symbolic machine learning., in: Proceedings of the 2016 Conference on Big Data from Space (BiDS'16). Presented at the Big Data from Space (BiDS'16), pp. 156–159. doi:10.2788/854791

## Other useful Resources

### GHSL project

<http://ghsl.jrc.ec.europa.eu>

External components and libraries:

### MATLAB Runtime

<http://uk.mathworks.com/products/compiler/mcr/>

### GDAL

<http://www.gdal.org/>

# ANNEX 1. Index of Parameters

blocksize .....	27
Cloud_codes .....	22
cloudmask_layer .....	22
cloudmask_res .....	22
cloudthresh .....	22
csl_lambda .....	27
csl_qlev .....	25
fusion .....	28
hist_stretch .....	26
lev_q_maxlev .....	25
lev_q_minlev .....	25
lrn_set_bu_val .....	22
lrn_set_datamask_val .....	22
lrn_set_name .....	21
lrn_set_refine_val .....	22
lrn_set_res .....	21
lrn_set_water_val .....	22
lum_bands .....	21
MaxSceneInGroupRapidEye .....	27
min_max .....	26
min_no_samples .....	24
morphological .....	23
ms_bands .....	24
ns .....	26
ns_ptx .....	29
ptx_fstd .....	26
ptx_RefineLearning .....	26
ptx_RefineOutput .....	23, 29
ptx_wsize .....	26, 27
rad_q_maxlev .....	24
rad_q_minlev .....	24
rad_qlev .....	24
sal_q_maxlev .....	25
sal_q_minlev .....	25
Sensor .....	21
UseBlockProc .....	27
UseParallelFlag .....	27
val_set_bu_val .....	23
val_set_datamask_val .....	23
val_set_name .....	22
val_set_res .....	23
vrt_input .....	21

# ANNEX 2. Format of parameters

## 1) Highlighted parameters

In the configuration file (.ods), some parameters are highlighted. They correspond to those parameters that are required with the present settings.

Whenever a parameter is modified, the required cells that need to be filled are dynamically highlighted.

## 2) The input parameters are of three data types: string, logical or numbers

### ○ String

If the parameter requires a string input (for example a filepath, type of operation or WMS layer name), the name should be typed without apostrophes or quotation marks.

#### Examples:

ID	Input	Parameter	Example
1.1	Filepath	lrn_set_name	P:\GHSL\learning.tif
1.2	Operation	fusion	intersection
1.3	WMS layer	LAYERS*	BU_learn

### ○ Logical

Logical (boolean) parameters marked as true or false

#### Examples:

ID	Input	Parameter	Example
2.1	Boolean	vrt_input	false

### ○ Numeric

Numeric parameters could be integers, float or vector values

#### Examples:

ID	Input	Parameter	Example
3.1	Integer	lrn_set_water_val	1
3.2	Float	sal_q_maxlev	0.95
3.3	Vector**	lrn_set_datamask_val	[1 2 3 4 6 8 9 10]
3.3.1	Vector**	lrn_set_datamask_val	[1 2 3 4 6 8 9 10]
3.3.2	Vector**	lrn_set_datamask_val	[1, 2, 3, 4, 6, 9, 10]
3.3.3	Vector**	lrn_set_datamask_val	[1:4 6 8:10]

Europe Direct is a service to help you find answers to your questions about the European Union  
Free phone number (\*): 00 800 6 7 8 9 10 11  
(\*) Certain mobile telephone operators do not allow access to 00 800 numbers or these calls may be billed.

A great deal of additional information on the European Union is available on the Internet.  
It can be accessed through the Europa server <http://europa.eu>

### How to obtain EU publications

Our publications are available from EU Bookshop (<http://bookshop.europa.eu>), where you can place an order with the sales agent of your choice.

The Publications Office has a worldwide network of sales agents. You can obtain their contact details by sending a fax to (352) 29 29-42758.

### JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.



**EU Science Hub**  
ec.europa.eu/jrc



@EU\_ScienceHub



EU Science Hub - Joint Research Centre



Joint Research Centre



EU Science Hub

